# UNIVERSITYOF BIRMINGHAM

# Low-CP-rank tensor completion via practical regularization

Jiang, Jiahua; Sanogo, Fatoumata; Navasca, Carmeliza

[Link to publication on Research at Birmingham portal](#)

Download date: 24. Apr. 2024

# Low-CP-Rank Tensor Completion via Practical Regularization

Jiahua Jiang[1] · Fatoumata Sanogo[2] · Carmeliza Navasca[2]

## Abstract

Dimension reduction is analytical methods for reconstructing high-order tensors that the intrinsic rank of these tensor data is relatively much smaller than the dimension of the ambient measurement space. Typically, this is the case for most real world datasets in signals, images and machine learning. The CANDECOMP/PARAFAC (CP, aka Canonical Polyadic) tensor completion is a widely used approach to find a low-rank approximation for a given tensor. In the tensor model (Sanogo and Navasca in 2018 52nd Asilomar conference on signals, systems, and computers, pp 845–849, https://doi.org/10.1109/ACSSC.2018.8645405, 2018), a sparse regularization minimization problem via $\ell_1$ norm was formulated with an appropriate choice of the regularization parameter. The choice of the regularization parameter is important in the approximation accuracy. Due to the emergence of the massive data, one is faced with an onerous computational burden for computing the regularization parameter via classical approaches (Gazzola and Sabaté Landman in GAMM-Mitteilungen 43:e202000017, 2020) such as the weighted generalized cross validation (WGCV) (Chung et al. in Electr Trans Numer Anal 28:2008, 2008), the unbiased predictive risk estimator (Stein in Ann Stat 9:1135–1151, 1981; Vogel in Computational methods for inverse problems, 2002), and the discrepancy principle (Morozov in Doklady Akademii Nauk, Russian Academy of Sciences, pp 510–512, 1966). In order to improve the efficiency of choosing the regularization parameter and leverage the accuracy of the CP tensor, we propose a new algorithm for tensor completion by embedding the flexible hybrid method (Gazzola in Flexible krylov methods for lp regularization) into the framework of the CP tensor. The main benefits of this method include incorporating the regularization automatically and efficiently as well as improving

✉ Jiahua Jiang
  j.jiang.3@bham.ac.uk

  Fatoumata Sanogo
  sanogof1@uab.edu

  Carmeliza Navasca
  cnavasca@uab.edu

[1]  School of Mathematics, University of Birmingham, Birmingham B15 2TS, UK

[2]  Department of Mathematics, University of Alabama at Birmingham, Birmingham, AL 35294, USA

🖄 Springer

accuracy in the reconstruction and algorithmic robustness. Numerical examples from image reconstruction and model order reduction demonstrate the efficacy of the proposed algorithm.

## 1 Introduction

Tensor computations have become prevalent in across many fields in mathematics [16, 28], computer science [9, 22, 35], engineering [14] and data science [1, 27]. In particular, tensor methods are now ubiquitous in areas of numerical linear algebra [6, 7], imaging sciences [21, 55] and applied algebraic geometry [29]. In addition, tensor based methods are gaining grounds in solving complex problems in scientific computing. [11].

The tensor rank problem is crucial in reconstructing a given tensor $\mathcal{T}$. The rank of tensor is defined as the minimum number $R$ of the summands in a sum of rank-one tensor product, $\sum_{r=1}^{R} \alpha_r \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r$. In practice, the column vectors, $\mathbf{a}_r$, $\mathbf{b}_r$ and $\mathbf{c}_r$, are concatenated into what we call factor matrices, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. The elements of the vector $\alpha$ of size $R$ are the scalings of the rank-one tensors. This tensor factorization is the well-known canonical polyadic or CANDECOMP/PARAFAC (CP) decomposition. Optimization techniques, namely the multi-block alternating methods are standard methods for finding the factor matrices of a given tensor and its rank.

A tensor model [45] which incorporates tensor rank approximation is

$$\min_{A,B,C,\alpha} \|\mathcal{T} - \mathcal{S}\|_F + \lambda\|\alpha\|_{\ell_1},$$

where $\lambda$ is a regularization parameter and $\mathcal{S} = \sum_{r=1}^{U} \alpha_r \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r$ with an upper bound tensor rank $R$. This sparse optimization problem is solved iteratively and the vector $\alpha$ reveals an approximated tensor rank $R$ where $R << U$. The main drawback of this model is that the accuracy of the computational results highly relies on an appropriate value of the regularization parameter $\lambda$. In [54], the choice of the regularization parameter is tied to two intrinsic parameters: variance of noise and incoherence of the given tensor data $\mathcal{T}$. One has to initialize $\lambda$ from a bound based on these two parameters from the tensor data with an upper bound rank $R$. In practice, a priori estimates of the variance and incoherence parameters are required based on a CP decomposition of the given data with an initial tensor rank guess $R$. Then $\lambda$ can be chosen accordingly from an estimated bound. The advantage of this approach is that it provides a theoretical bound for $\lambda$. However, it is not practical enough for real data implementation; the choice of $\lambda$ is only as good as the estimated intrinsic parameter. Moreover, the accuracy level is only around $10^{-2}$. In this paper, we present a more adaptive, practical and methodical way for calculating the regularization parameter $\lambda$ using the flexible hybrid method, which is tailored for the use in the canonical CP tensor framework. The flexible hybrid method is capable of computing the regularized solutions to large-scale linear inverse problems more efficiently than classical approaches [20], since the original regularized problem is iteratively projected onto small subspaces of increasing dimension and the regularization parameter $\lambda$ is estimated by implementing the standard regularization techniques, such as weighted generalized cross validation (WGCV) [15], unbiased predictive risk estimator [48, 51], and discrepancy principle [36], on the projected problem at each iteration. Our numerical results show that this new iterativedt8;333 'method gives more accurate results in tensor completion and model order reduction problems.

We have two application areas in this paper: tensor completion in image restoration and model order reduction. Matrix and tensor completion techniques provide major tools in recommender systems in computer science and in general in data science; it is about filling in missing entries from the partially observed entries of the matrix or tensor. The success of matrix completion methods are attributed to sparse optimization methods in compressed sensing [13]. These methods have been generalized iteratively to tensor completion problem [33] via Tucker models [4, 52] where missing entries are predicted through the matrix trace norm optimization. In fact, the tensor completion problem dates back as early as in 2000. Bro [12] had one of the earliest work on demonstrating two ways to handle missing data using CP. The first one is to estimate the model parameters iteratively while imputing the missing data. Another approach called Missing-Skipping skips the missing value and builds up the model based only on the observed part via a weighted least squares formulation in the CP format [2]. Our proposed tensor completion gives more accurate results in capturing more features in color images through low rank construction via our model and numerical technique.

Furthermore, we show that recent effort in tensor-based model reduction such as Randomized CP tensor decomposition [17] and tensor POD [56] have been rewarded with many promising developments leverage the computational effort for many-query computations and repeated output evaluations for different values of some inputs of interest where classical model order reduction approaches [38, 39] such as Reduced Basis Methods [8, 44] and Proper Orthogonal Decomposition (POD) faced with heavy computational burden. Compared with the classical model order reduction approaches, tensor-based model reduction algorithms allow us to achieve significant computational savings, especially for expensive high fidelity numerical solvers.

The paper is organized as follows. In Sect. 2, we provide some tensor backgrounds and basic tensor decomposition. Then, Sect. 3 deals with the derivations of the iterative equation by alternating block optimization containing the unfolding of the tensors in each mode. The proximal gradient formulation and the flexible hybrid method for the automatic selection of the regularization parameter are discussed. Experimental results are in Sect. 4, and the conclusions follow in Sect. 5.

## 2 Preliminaries

We denote a vector by a bold lower-case letter $\mathbf{a}$. The bold upper-case letter $\mathbf{A}$ represents a matrix and the symbol of tensor is a calligraphic letter $\mathcal{A}$. Throughout this paper, we focus on third-order tensors $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{I \times J \times K}$ of three indices $1 \leq i \leq I$, $1 \leq j \leq J$ and $1 \leq k \leq K$, but all are applicable to tensors of arbitrary order greater or equal to three.

A third-order tensor $\mathcal{A}$ has column, row and tube fibers, which are defined by fixing every index but one and denoted by $\mathbf{a}_{:jk}$, $\mathbf{a}_{i:k}$ and $\mathbf{a}_{ij:}$ respectively. Correspondingly, we can obtain three kinds $\mathbf{A}_{(1)}$, $\mathbf{A}_{(2)}$ and $\mathbf{A}_{(3)}$ of matricization of $\mathcal{A}$ according to respectively arranging the column, row, and tube fibers to be columns of matrices. We can also consider the vectorization for $\mathcal{A}$ to obtain a row vector $\mathbf{a}$ such the elements of $\mathcal{A}$ are arranged according to $k$ varying faster than $j$ and $j$ varying faster than $i$, i.e., $\mathbf{a} = (a_{111}, \ldots, a_{11K}, a_{121}, \ldots, a_{12K}, \ldots, a_{1J1}, \ldots, a_{1JK}, \ldots)$.

Henceforth, the outer product of a rank-one third order tensor is denoted as $\mathbf{x} \circ \mathbf{y} \circ \mathbf{z} \in \mathbb{R}^{I \times J \times K}$ of three nonzero vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ is a rank-one tensor with elements $x_i y_j z_k$ for all the indices. A canonical polyadic decomposition of $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ expresses $\mathcal{A}$ as a sum of rank-one outer products:

$$\mathcal{A} = \sum_{r=1}^{R} \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r \tag{2.1}$$

where $\mathbf{x}_r \in \mathbb{R}^I$, $\mathbf{y}_r \in \mathbb{R}^J$, $\mathbf{z}_r \in \mathbb{R}^K$ for $1 \leq r \leq R$. Every outer product $\mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r$ is called as a rank-one component and the integer $R$ is the number of rank-one components in tensor $\mathcal{A}$. The minimal number $R$ such that the decomposition (2.1) holds is the rank of tensor $\mathcal{A}$, which is denoted by rank($\mathcal{A}$). For any tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$, rank($\mathcal{A}$) has an upper bound $\min\{IJ, JK, IK\}$ [30].

In this paper, we consider CP decomposition in the following form

$$\mathcal{T} = \sum_{r=1}^{R} \alpha_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{2.2}$$

where $\alpha_r \in \mathbb{R}$ is a rescaling coefficient of rank-one tensor $\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ for $r = 1, \ldots, R$. For convenience, we let $\alpha = (\alpha_1, \ldots, \alpha_R) \in \mathbb{R}^R$ and denote $[\alpha; \mathbf{A}, \mathbf{B}, \mathbf{C}]_R = \sum_{r=1}^{R} \alpha_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ in (2.2) where $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_R) \in \mathbb{R}^{I \times R}$, $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_R) \in \mathbb{R}^{J \times R}$ and $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_R) \in \mathbb{R}^{K \times R}$ are called the factor matrices of tensor $\mathcal{A}$.

In most iterative techniques for tensor decompositions, the high order tensor *matricizations* are transformed from tensor equations into matrix equations via the standard unfolding mechanishm. Here we describe a standard approach for a matricizing of a tensor. The Khatri-Rao product [47] of two matrices $X \in \mathbb{R}^{I \times R}$ and $Y \in \mathbb{R}^{J \times R}$ is defined as

$$X \odot Y = (x_1 \otimes y_1, \ldots, x_R \otimes y_R) \in R^{IJ \times R},$$

where the symbol "$\otimes$" denotes the Kronecker product:

$$x \otimes y = (x_1 y_1, \ldots, x_1 y_J, \ldots, x_I y_1, \ldots, x_I y_J)^T.$$

Using the Khatri-Rao product, the CP model (2.2) can be written in three equivalent matrix equations:

$$\mathbf{T}_{(1)} = \mathbf{A}\mathrm{Diag}(\mathbf{C} \odot \mathbf{B})^T, \tag{2.3a}$$

$$\mathbf{T}_{(2)} = \mathbf{B}\mathrm{Diag}(\mathbf{C} \odot \mathbf{A})^T, \tag{2.3b}$$

and

$$\mathbf{T}_{(3)} = \mathbf{C}\mathrm{Diag}(\mathbf{B} \odot \mathbf{A})^T \tag{2.3c}$$

where **Diag** is a diagonal matrix, where the diagonal entries are the elements of $\alpha$. To achieve CP decomposition of given tensor $\mathcal{T}$ with a known tensor rank $R$ and an assumption that $\mathbf{D} = \mathbf{I}$, the matrix equations Eqs. (2.3a)–(2.3c) are formulated into linear least-squares subproblems to solve iteratively for $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, respectively. Here are the linear least-squares subproblems:

$$\min_{\mathbf{A}} \|\mathbf{T}_{(1)} - \mathbf{A}\mathrm{Diag}(\mathbf{C} \odot \mathbf{B})^T\|_F^2, \tag{2.4a}$$

$$\min_{\mathbf{B}} \|\mathbf{T}_{(2)} - \mathbf{B}\mathrm{Diag}(\mathbf{C} \odot \mathbf{A})^T\|_F^2, \tag{2.4b}$$

and

$$\min_{\mathbf{C}} \|\mathbf{T}_{(3)} - \mathbf{C}\mathrm{Diag}(\mathbf{B} \odot \mathbf{A})^T\|_F^2. \tag{2.4c}$$

This technique is the well known Alternating Least-Squares (ALS) [12, 31]. Typically, a normalization constraint on factor matrices such that each column is normalized to length one [3, 50] is required for convergence, which we denote by $\mathbf{N}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = 1$.

## 3 Iterative Equations for Tensor Completion

We will describe our low rank tensor model of a given tensor in a CP format with an approximated tensor rank for tensor completion. Our goal is to fill in the missing entries from a given tensor $\mathcal{T}$ with the partially observed entries by reconstructing a completed low rank tensor $\mathcal{S}$. To do so, we formulate a sparse optimization problem [45] for recovering CP decomposition from tensor $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$ with partially observed entries on the index set $\Omega$:

$$\min_{A,B,C,\alpha} \|\mathcal{T} - \mathcal{S}\|_F + \lambda \|\alpha\|_{\ell_1}$$

$$\text{subject to } \mathcal{S}(\Omega) = \mathcal{T}(\Omega) \tag{3.1a}$$

where $\lambda$ is a constant regularization parameter and $\mathcal{S} = \sum_r \alpha_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$.

We will now derive the iterative equations for $A$, $B$, $C$ and $\alpha$. The equations are typically associated with Iterative Soft Thresholding Algorithm (ISTA) [5] whose derivation is based on the Majorization-Minimization (MM) [18] method. ISTA (Iterative Soft-Thresholding Algorithm) is a combination of the Landweber algorithm and soft-thresholding (so it is also called the thresholded-Landweber algorithm).

Suppose we have a minimization problem:

$$\min_x f(x). \tag{3.2}$$

By using the proximal operators formulation (see Appendix) and the MM approach, we first find an upper bound for $f(x)$:

$$f(x) \leq f(y) + \langle \nabla_x f(y), x - y \rangle + \gamma \|x - y\|_2^2$$

Let $g(x, y) = f(y) + \langle \nabla_x f(y), x - y \rangle + \gamma \|x - y\|_2^2$. Note that $f(x) \leq g(x, y)$ for all $x$ and $f(x) = g(x, y)$ when $y = x$. Thus, we can reformulate 3.2 as

$$\min_x f(y) + \langle \nabla_x f(y), x - y \rangle + \gamma \|x - y\|_2^2. \tag{3.3}$$

Since this is a minimization over $x$, then 3.3 is equivalent to

$$\min_x \langle \nabla_x f(y), x - y \rangle + \gamma \|x - y\|_2^2. \tag{3.4}$$

By gathering the terms with respect to $x$, the objective function in 3.4 can be expressed as

$$\gamma(-2b^T x + x^T x) + c,$$

where $c = \gamma(y)^T y - \nabla_x f(y)^T y$ and $b = y - \frac{1}{2\gamma} \nabla_x f(y)$. Since $b^T b - 2b^T x + x^T x = \|b - x\|_2^2$, we have a new formulation:

$$\min_x \gamma \|x - b\|_2^2. \tag{3.5}$$

Now from the least-squares problems (2.4a–2.4c) and using proximal gradient formulation, we have the following new formulations:

$$\mathbf{A}^* = \arg\min_{\mathbf{A}} \{\langle \mathbf{A} - \mathbf{A}^n, \nabla_\mathbf{A} f(\mathbf{A}^n, \mathbf{B}^n, \mathbf{C}^n, \alpha^n) \rangle + \frac{sd_n}{2} \|\mathbf{A} - \mathbf{A}^n\|_F^2\}$$

$$\text{s.t. } \|\mathbf{a}_i\| = 1, i = 1, \dots, R,$$

$$\mathbf{B}^* = \arg\min_{\mathbf{B}} \{\langle \mathbf{B} - \mathbf{B}^n, \nabla_\mathbf{B} f(\mathbf{A}^{n+1}, \mathbf{B}^n, \mathbf{C}^n, \alpha^n) \rangle + \frac{se_n}{2} \|\mathbf{B} - \mathbf{B}^n\|_F^2\}$$

$$\text{s.t. } \|\mathbf{b}_i\| = 1, i = 1, \dots, R,$$

and

$$\mathbf{C}^* = \arg\min_{\mathbf{C}}\{\langle \mathbf{C} - \mathbf{C}^n, \nabla_{\mathbf{C}} f(\mathbf{A}^{n+1}, \mathbf{B}^{n+1}, \mathbf{C}^n, \alpha^n)\rangle + \frac{sf_n}{2}\|\mathbf{C} - \mathbf{C}^n\|_F^2\}$$

$$\text{s.t. } \|\mathbf{c}_i\| = 1, i = 1, \dots, R,$$

where $d_n = \max\{\|\mathbf{U}^n \mathbf{U}^{nT}\|_F, 1\}$, $e_n = \max\{\|\mathbf{V}^n \mathbf{V}^{nT}\|_F, 1\}$ and $f_n = \max\{\|\mathbf{W}^n \mathbf{W}^{nT}\|_F, 1\}$ with

$$\mathbf{U} = \mathbf{Diag}(\mathbf{C} \odot \mathbf{B})^T, \mathbf{V} = \mathbf{Diag}(\mathbf{C} \odot \mathbf{A}^T), \mathbf{W} = \mathbf{Diag}(\mathbf{B} \odot \mathbf{A})^T. \quad (3.6)$$

The gradients of $f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \alpha) = \frac{1}{2}\|\mathcal{T} - \sum_{r=1}^{R} \alpha_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\|_F^2$ on $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are the following in terms of the Khatri-Rao product via matricizations:

$$\nabla_{\mathbf{A}} f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \alpha) = (\mathbf{A}\mathbf{Diag}(\mathbf{C} \odot \mathbf{B}) - \mathbf{T}_{(1)})(\mathbf{C} \odot \mathbf{B})^T \mathbf{Diag}^T, \quad (3.7a)$$

$$\nabla_{\mathbf{B}} f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \alpha) = (\mathbf{B}\mathbf{Diag}(\mathbf{C} \odot \mathbf{A}) - \mathbf{T}_{(2)})(\mathbf{C} \odot \mathbf{A})^T \mathbf{Diag}^T, \quad (3.7b)$$

and

$$\nabla_{\mathbf{C}} f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \alpha) = (\mathbf{C}\mathbf{Diag}(\mathbf{B} \odot \mathbf{A}) - \mathbf{T}_{(3)})(\mathbf{B} \odot \mathbf{A})^T \mathbf{Diag}^T. \quad (3.7c)$$

The gradient of $f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \alpha)$ on $\alpha$ is described in Sect. 3.1. Based from the calculations (3.3–3.5), we obtain the following iterative formula for $A$:

$$\arg\min_{\mathbf{A}}\{\|\mathbf{A} - \mathbf{D}^n\|_F^2\} \quad \text{s.t. } \|\mathbf{a}_i\| = 1, i = 1, \dots, R.$$

where $\mathbf{D}^n = \mathbf{A}^n - \frac{1}{sd_n}\nabla_{\mathbf{A}} f(\mathbf{A}^n, \mathbf{B}^n, \mathbf{C}^n, \alpha^n)$. We can break it further component-wise:

$$\mathbf{a}_i^{n+1} = \mathbf{d}_i^n / \|\mathbf{d}_i^n\|, i = 1, \dots, R,$$

where $\mathbf{a}_i^{n+1}$ and $\mathbf{d}_i^n$ are the $i$-th columns of $\mathbf{A}^{n+1}$ and $\mathbf{D}^n$.
Similarly, the update of $\mathbf{B}$ is

$$\arg\min_{\mathbf{Y}}\{\|\mathbf{B} - \mathbf{E}^n\|_F^2\} \quad \text{s.t. } \|\mathbf{b}_i\| = 1, i = 1, \dots, R.$$

where $\mathbf{E}^n = \mathbf{B}^n - \frac{1}{se_n}\nabla_{\mathbf{B}} f(\mathbf{A}^{n+1}, \mathbf{B}^n, \mathbf{C}^n, \alpha^n)$. Column-wise, we have

$$\mathbf{b}_i^{n+1} = \mathbf{e}_i^n / \|\mathbf{e}_i^n\|, i = 1, \dots, R,$$

where $\mathbf{b}_i^{n+1}$ and $\mathbf{e}_i^n$ are the $i$-th columns of $\mathbf{B}^{n+1}$ and $\mathbf{E}^n$.
Furthermore, the update of $\mathbf{C}$ is

$$\arg\min_{\mathbf{C}}\{\|\mathbf{C} - \mathbf{F}^n\|_F^2\} \quad \text{s.t. } \|\mathbf{c}_i\| = 1, i = 1, \dots, R.$$

where $\mathbf{F}^n = \mathbf{C}^n - \frac{1}{sf_n}\nabla_{\mathbf{C}} f(\mathbf{A}^{n+1}, \mathbf{B}^{n+1}, \mathbf{C}^n, \alpha^n)$. Also, we update vector-wise:

$$\mathbf{c}_i^{n+1} = \mathbf{f}_i^n / \|\mathbf{f}_i^n\|, i = 1, \dots, R,$$

where $\mathbf{c}_i^{n+1}$ and $\mathbf{f}_i^n$ are the $i$-th columns of $\mathbf{C}^{n+1}$ and $\mathbf{F}^n$.

### 3.1 Iterative Equation for $\alpha$

Using the vectorization of tensors in Sect. 2, we can vectorize every rank-one tensor of outer product $\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ into a row vector $\mathbf{q}_r$ for $1 \leq r \leq R$. We denote a matrix consisting of all $\mathbf{q}_r$ for $1 \leq r \leq R$ by

$$\mathbf{Q} = (\mathbf{q}_1^T, \ldots, \mathbf{q}_R^T)^T. \tag{3.8}$$

Thus the function $\frac{1}{2}\|\mathcal{T} - \sum_{r=1}^R \alpha_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\|_F^2$ can be also written as $\frac{1}{2}\|\mathbf{t} - \alpha\mathbf{Q}\|_F^2$ where $\mathbf{t}$ is a vectorization for tensor $\mathcal{T}$. Also, the gradient of $f(\bullet)$ on $\mathbf{A}, \mathbf{B}, \mathbf{C}$, is the following in terms of the Khatri-Rao product via matricizations:

$$\nabla_\alpha f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \alpha) = (\alpha\mathbf{Q} - \mathbf{t})\mathbf{Q}^T.$$

Then, the minimization problem for $\alpha$ is

$$\min_\alpha \frac{1}{2}\|\mathbf{t} - \alpha\mathbf{Q}\|_F^2 + \lambda\|\alpha\|_1. \tag{3.9}$$

Efficiently and appropriately choosing the regularization parameter $\lambda$ plays a crucial role in solving (3.9). In the papers [53, 54], the proximal operators formulation (see Appendix) and the MM approach are used to solve $\alpha$ iteratively via

$$\alpha^{n+1} = \arg\min_\alpha \{\langle \alpha - \alpha^n, \nabla_\alpha f(\mathbf{A}^{n+1}, \mathbf{B}^{n+1}, \mathbf{C}^{n+1}, \alpha^n)\rangle + \frac{s\eta_n}{2}\|\alpha - \alpha^n\|^2 + \lambda\|\alpha\|_1\}.$$

which is equivalent to the following:

$$\alpha^{n+1} = \arg\min_\alpha \frac{1}{2}\left\|\alpha - \alpha^n + \frac{1}{s\eta_n}\nabla_\alpha f(\mathbf{A}^{n+1}, \mathbf{B}^{n+1}, \mathbf{C}^{n+1}, \alpha^n)\right\|^2 + \frac{\lambda}{s\eta_n}\|\alpha\|_1 \tag{3.10}$$

where $\eta_n = \max\{\|\mathbf{Q}^{n+1}\mathbf{Q}^{n+1T}\|_F, 1\}$.

However, we found that the accuracy of these methods heavily depends on the choice of the initial value of $\alpha$, which reduces the robustness of the entire algorithm, especially for practical problems. To address this problem, we embed the flexible hybrid method introduced in the following section into the CP completion framework.

### 3.2 The Flexible Hybrid Method for $\ell_1$ Regularization

Efficiently and accurately solving (3.9) is important for the CP completion. Notice that (3.9) is an inverse problem with $\ell_1$ regularization. The iteratively reweighted norm (IRN) methods [24, 43] are typical strategies for solving the $\ell_p-$ regularization inverse problem. However, these methods assume that an appropriate value of the regularization parameter is known in advance, which is hard oftentimes. Therefore, there have been some recent works [19, 23] on selecting regularization parameters for $\ell_p$. In this work, we focus on $p = 1$ and employing the flexible method based on Golub–Kahan process [19] to solve the $\ell_1-$regularized problem. For the convenience of description, we rewrite problem (3.9) into a standard inverse problem with $\ell_1$ regularization as follows.

$$\min_s \|\mathbf{H}s - \mathbf{d}\|_2^2 + \lambda\|s\|_1, \tag{3.11}$$

where $\mathbf{d} \in \mathbb{R}^m$ is the observed data, $\mathbf{H} \in \mathbb{R}^{m \times n}$ models the forward process, $s \in \mathbb{R}^n$ is the approximation of the desired solution. The first step is to break the $\ell_1-$regularized problem (3.11) into a sequence of $\ell_2-$norm problems,

$$\min_{s}||\boldsymbol{Hs} - \boldsymbol{d}||_2^2 + \lambda||\boldsymbol{L(s)s}||_2^2, \tag{3.12}$$

where

$$\boldsymbol{L(s)} = \text{diag}\left(\left(1/\sqrt{f_\tau([|\boldsymbol{s}|]_i)}\right)_{i=1,\dots,n}\right), \tag{3.13}$$

and $f_\tau([|\boldsymbol{s}|]_i) = \begin{cases} [|\boldsymbol{s}|]_i & [|\boldsymbol{s}|]_i \geq \tau_1 \\ \tau_2 & [|\boldsymbol{s}|]_i < \tau_1 \end{cases}$. Here $0 < \tau_2 < \tau_1$ are small thresholds enforcing some additional sparsity in $f_\tau([|\boldsymbol{s}|]_i)$. Since directly solving (3.12) is not possible in real problems since the true $\boldsymbol{s}$ is not available. To avoid nonlinearities and follow the common practice of iterative methods, $\boldsymbol{L(s)}$ can be approximated by $\boldsymbol{L(s_k)}$, where $\boldsymbol{s_k}$ is the numerical solution at the $(k-1)$-th iteration that can be treated as an approximation of the solution at $k$-th iteration. Since directly choosing regularization parameters for large problems is quite costly, the flexible hybrid approaches based on the flexible Golub–Kahan process[19] has been developed to solve the following variable-preconditioned Tikhonov problem,

$$\min_{s}||\boldsymbol{Hs} - \boldsymbol{d}||_2^2 + \lambda||\boldsymbol{L_k s}||_2^2, \tag{3.14}$$

which is equivalent to

$$\min_{s}||\boldsymbol{HL_k^{-1}}\widehat{\boldsymbol{s}} - \boldsymbol{d}||_2^2 + \lambda||\widehat{\boldsymbol{s}}||_2^2, \tag{3.15}$$

where $\widehat{\boldsymbol{s}} = \boldsymbol{L_k s}$, and $\boldsymbol{L_k} = \boldsymbol{L(s_k)}$ may change at each iteration. To be able to incorporate the changing preconditioner, the flexible Golub–Kahan process (FGK) is used to generate the bases for the solution. Given $\boldsymbol{H}, \boldsymbol{d}$ and changing preconditioner $\boldsymbol{L_k}$, the FGK iterative process can be described as follows. Let $\boldsymbol{u_1} = \boldsymbol{d}/\|\boldsymbol{d}\|_2$ and $\boldsymbol{v_1} = \boldsymbol{H}^\top \boldsymbol{u_1}/\|\boldsymbol{H}^\top \boldsymbol{u_1}\|_2$. Then at the $k$-th iteration, we generates vectors $\boldsymbol{p_k}, \boldsymbol{v_k}$ and $\boldsymbol{u_{k+1}}$ such that

$$\boldsymbol{HP_k} = \boldsymbol{U_{k+1}M_k} \quad \text{an}d \quad \boldsymbol{H}^\top \boldsymbol{U_{k+1}} = \boldsymbol{V_{k+1}T_{k+1}}, \tag{3.16}$$

where $\boldsymbol{P_k} = \begin{bmatrix} \boldsymbol{L_1^{-1}v_1} \dots \boldsymbol{L_k^{-1}v_k} \end{bmatrix} \in \mathbb{R}^{n \times k}$, $\boldsymbol{M_k} \in \mathbb{R}^{(k+1) \times k}$ is upper Hessenberg, $\boldsymbol{T_{k+1}} \in \mathbb{R}^{(k+1) \times (k+1)}$ is upper triangular, and $\boldsymbol{U_{k+1}} = \begin{bmatrix} \boldsymbol{u_1} \dots \boldsymbol{u_{k+1}} \end{bmatrix} \in \mathbb{R}^{m \times (k+1)}$ and $\boldsymbol{V_{k+1}} = \begin{bmatrix} \boldsymbol{v_1} \dots \boldsymbol{v_{k+1}} \end{bmatrix} \in \mathbb{R}^{n \times (k+1)}$ contain orthonormal columns. We remark that the column vectors of $\boldsymbol{P_k}$ don't span a Krylov subspace like conventional Golub–Kahan bidiagonalization process[10, 40], but they do provide a basis for the solution $\boldsymbol{s_k}$ at $k$-th iteration. Given the relationships in (3.16), an approximate least-squares solution can be computed as $\boldsymbol{s_k} = \boldsymbol{P_k q_k}$, where $\boldsymbol{q_k}$ is the solution to the projected least-squares problem,

$$\min_{\boldsymbol{s_k} \in \mathcal{R}(\boldsymbol{P_k})}||\boldsymbol{Hs_k} - \boldsymbol{d}||_2^2 = \min_{\boldsymbol{q_k}}||\boldsymbol{M_k q_k} - \beta_1 \boldsymbol{e_1}||_2^2, \tag{3.17}$$

where $\mathcal{R}(\boldsymbol{P_k})$ denotes the range of $\boldsymbol{P_k}$, $\boldsymbol{e_1} \in \mathbb{R}^{k+1}$ is the first column of $(k+1)$ by $(k+1)$ identity matrix. However, it is well-known that, for inverse problems, standard iterative methods, like LSQR, exhibit a semiconvergent behavior, whereby the solution improve in early iterations but become contaminated with inverted noise in later iterations [25]. This phenomennon, which is common for most ill-posed inverse problems, occurs also for the flexible methods. Thus, it is desirable to consider including a standard regularization term on the projected problem (3.17) to stabilize the reconstruction errors, so that

$$\boldsymbol{q_k} = \arg\min_{\boldsymbol{q}}||\boldsymbol{M_k q} - \beta_1 \boldsymbol{e_1}||_2^2 + \lambda||\boldsymbol{q}||_2^2. \tag{3.18}$$

Henceforth, $\boldsymbol{s_k} = \boldsymbol{P_k q_k}$ is the numerical solution at $k$-th iteration for the full problem. To get a better regularized solution, we consider using weighted generalized crossed validation (WGCV) method [15] to choose $\lambda$.

### 3.3 Practical Regularization CP Tensor

Recall that the low-rank CP tensor completion requiring solving $\alpha$ in (3.9) each iteration, can be computationally inaccurate and unstable since the forward matrix **Q** is ill-posed and that the regularization parameter $\lambda$ is hard to determine without enough prior knowledge. Our proposed *practial regularization CP tensor method* implements the flexible hybrid method in solving (3.9), which thus ameliorates the instability and improves the solution accuracy without sacrificing too much in speed by using standard parameter selection techniques such as WGCV to efficiently and automatically estimate the regularization parameter for the projected problem during the iterative process.

Notice that in practical regularization CP tensor method, iteratively updating $\alpha$ and $\lambda$ via the flexible hybrid method is independent of their initial settings. Furthermore, the flexible hybrid method has proven to be very effective for solving large linear inverse problems due to its inherent regularization properties as well as the added flexibility to select regularization parameters adaptively. Thus, compared with classical CP tensor, our proposed algorithm is more accurate and stable without sacrificing much efficiency. A summary pseudocode of the practical regularization CP tensor is provided in Algorithm 3.1, where $diag(a_1, \ldots, a_n)$ in the pseudocode denotes the diagonal matrix with the diagonal entries $a_1, \ldots, a_n$.

## 4 Numerical Results

In this section, we consider various scenarios where our proposed algorithm can enhance the stability and accuracy compared with the classical CP tensor method. In all the simulations, the initial guesses are randomly generated. The stopping criterion used in all experiments depends on two parameters: one is the upper bound of the number of iterations $m_{\max}$ and the other is the tolerance $\epsilon_{\text{tol}}$ of the relative difference between the observation and the approximation. The regularization parameter $\lambda$ is iteratively updated by the flexible Krylov method with weight generalized cross validation method. These experiments ran on a laptop computer with Intel i5 CPU 2GHz and 16G memory.

### 4.1 Image Recovering by Tensor Completion

For the first experiment, we test two cases for this example, where the missing pixels for the first case are randomly chosen while the miss part for the second case is deterministic. The reconstruction error is computed with the relative error $\frac{||\mathcal{A}_n - \mathcal{A}||_2}{||\mathcal{A}||_2}$, where $\mathcal{A}_n$ denotes the approximated tensor at the $n$th iteration, and $\mathcal{A}$ represents the true tensor we want to reconstruct.

**Case 1** We implemented our algorithm on a color image $\mathcal{A} \in \mathbb{R}^{189 \times 267 \times 3}$ shown in Fig. 1. We recovered an estimated color image after removing 30% of the entries from the origin color image, which is shown in Fig. 1. The upper bound $R$ of rank($\mathcal{A}$) is fixed to 50 in the algorithm. The stopping criteria for this case are assumed to be $m_{\max} = 500$ and the relative error tolerance is $\epsilon_{\text{tol}} = 10^{-3}$.

The recovered images by original CP tensor with different choices of $\lambda$ and practical regularization CP tensor are provided in Fig. 2. The results of these two approaches both correspond to 500 iterations. It is noted that, for the conventional CP tensor, the quality of image recovery highly depends on the choice of the regularization parameter $\lambda$. Our proposed practical regularization CP tensor produces recovered image that has much less noise than

---

**Algorithm 3.1** Practical regularization CP tensor

---

**Input:** A third order tensor $\mathcal{A}$, an upper bound $R$ of rank($\mathcal{A}$), a penalty parameter $\lambda$ and a scale $s > 1$;
**Output:** An approximated tensor $\widetilde{\mathcal{B}}$ with an estimated rank $\widehat{R}$.

1. Give an initial tensor $\mathcal{B}^0 = [\alpha^0; \mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0]_R$.
2. Update step :
    (a) Update matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$:
        i. Compute $U^n$ by (3.6) and let $d_n = \max\{\left\|U^n U^{n\top}\right\|_F, 1\}$.
        ii. Compute $\mathbf{D}^n$ and $\mathbf{A}^{n+1}$ by

$$\mathbf{D}^n = \mathbf{A}^n - \frac{1}{sd_n}\nabla_\mathbf{A} f(\mathbf{A}^n, \mathbf{B}^n, \mathbf{C}^n, \alpha^n),$$

$$\mathbf{A}^{n+1} = \mathbf{D}^n diag(\left\|d_1^n\right\|_2, \ldots, \left\|d_R^n\right\|_2)^{-1},$$

   where $d_i^n$ is the $i$−th column of $\mathbf{D}^n$ for $i = 1, \ldots, R$.
        iii. Compute $V^n$ by (3.6) and let $e_n = \max\{\left\|V^n V^{n\top}\right\|_F, 1\}$.
        iv. Compute $\mathbf{E}^n$ and $\mathbf{B}^{n+1}$ by

$$\mathbf{E}^n = \mathbf{B}^n - \frac{1}{se_n}\nabla_\mathbf{B} f(\mathbf{A}^{n+1}, \mathbf{B}^n, \mathbf{C}^n, \alpha^n),$$

$$\mathbf{B}^{n+1} = \mathbf{E}^n diag(\left\|e_1^n\right\|_2, \ldots, \left\|e_R^n\right\|_2)^{-1},$$

   where $e_i^n$ is the $i$−th column of $\mathbf{E}^n$ for $i = 1, \ldots, R$.
        v. Compute $\mathbf{W}^n$ by (3.6) and let $f_n = \max\{\left\|\mathbf{W}^n \mathbf{W}^{n\top}\right\|_F, 1\}$.
        vi. Compute $\mathbf{F}^n$ and $\mathbf{C}^{n+1}$ by

$$\mathbf{F}^n = \mathbf{C}^n - \frac{1}{sf_n}\nabla_\mathbf{C} f(\mathbf{A}^{n+1}, \mathbf{B}^{n+1}, \mathbf{C}^n, \alpha^n),$$

$$\mathbf{C}^{n+1} = \mathbf{F}^n diag(\left\|\mathbf{f}_1^n\right\|_2, \ldots, \left\|\mathbf{f}_R^n\right\|_2)^{-1},$$

   where $\mathbf{f}_i^n$ is the $i$−th column of $\mathbf{F}^n$ for $i = 1, \ldots, R$.
    (b) Update the row vector $\alpha$:
        i. Compute $\mathbf{Q}^{n+1}$ by (3.8) and let $\eta_n = \max\{\|\mathbf{Q}^{n+1}\mathbf{Q}^{n+1^T}\|_F, 1\}$.
        ii. Solve (3.9) for $\alpha$ by the flexible hybrid method introduced in Section 3.2 and let

$$\alpha^{n+1} = \alpha.$$

3. Denote the limitations by $\widehat{A}, \widehat{B}, \widehat{C}, \widehat{\alpha}$, compute $\widehat{\mathcal{B}} = [\widehat{\alpha}; \widehat{A}, \widehat{B}, \widehat{C}]_R$ and count the number $\widehat{R}$ of nonzero entries in $\widehat{\alpha}$.
4. Impose constraints $\mathcal{A}(\Omega) = \mathcal{B}(\Omega)$, if the information of missing entries of $\mathcal{A}$ is available and that $\Omega$ denotes all the non-missing entries.
5. **return** The tensor $\widehat{\mathcal{B}}$ with the estimated rank $\widehat{R}$.

---

classical CP tensor, demonstrating that using flexible Krylov method to determine different regularization parameter for each iteration is beneficial. The comparison of the relative error shown in Fig. 2 also verifies the better performance of our practical regularization CP tensor. In terms of CPU time, original CP tensor with $\lambda = 35$ took 14.1 sec, and practical regularization CP tensor took 43.4 sec, since the automatic regularization parameter selection by flexible Krylov method is more expensive than predetermined regularization parameter that is cost-free.

**Case 2** We consider recovering the image $\mathcal{A} \in \mathbb{R}^{246 \times 257 \times 3}$ with the certain missing pixels as shown in Fig. 3, associating with its true image. The upper bound $R$ of rank ($\mathcal{A}$) is chosen

**Fig. 1** True image (left) and observation (right)

to be 50. The stopping criteria are setup as $m_{\max} = 250$ and the relative error tolerance is $\epsilon_{\text{tol}} = 10^{-3}$. For the classical CP tensor, we choose $\lambda = 35$.

The recovered images are provided in Fig. 4. The original CP tensor reached the stopping criteria at 12th iteration, and took 0.7 s for the quick convergence. For our proposed practical regularization CP tensor, the maximum number of iteration was reached, and the running time of it is 26.1 s. We observe that our algorithm provide more complete recovery than conventional CP tensor, which is also demonstrated by the relative error. To avoid the premature convergence problem, the results of original CP tensor corresponding to the maximum iterations has been checked, and is similar to the results at 12th iteration.

## 4.2 Model Order Reduction

Next we investigate a scenario in model order reduction where the key snapshots needs to be obtained to capture the low rank structure of the solution manifold that has low Kolmogorov width [34, 42]. This example demonstrates some advantages of our practical regularization CP tensor. Model order reduction techniques such as the POD and the Reduced Basis Methods are typically used to solve the problems requiring one to query an expensive yet deterministic computational solver once for each parameter node. We show that hybridizing our approach with the regularized alternating block minimization method [32, 37] provides a novel way to do model reduction and pattern extraction. More specifically, assuming $\mathcal{A}$ is the collection of the solutions on sampled parameters. To select the snapshots (reduced bases) for the low rank approximation of the solution manifold, we employ our algorithm to give a prior knowledge of rank$(\mathcal{A})$ denoted as $R$. Then we run the regularized alternating block minimization method according to the $R$ to approximate $\mathcal{A}$ and construct the reduced bases, where the stopping criteria for the outer ALS step are also defined by the maximum number of the iteration, which we assume here to be 500, and a tolerance of the relative error $\frac{||\mathcal{A}_n - \mathcal{A}||_2}{||\mathcal{A}||_2}$, which we assume to be $\epsilon_{\text{tol}} = 10^{-2}$.

In this experiment, we consider the following two-dimensional diffusion equation that induces a solution manifold that requires many snapshots to achieve small error:

$$(1 + \mu_1 x)u_{xx} + (1 + \mu_2 y)u_{yy} = e^{4xy} \quad \text{on} \quad \Omega. \tag{4.1}$$

The physical domain is $\Omega = [-1, 1] \times [-1, 1]$ and we impose homogeneous Dirichlet boundary conditions on $\partial\Omega$. The truth approximation is a spectral Chebyshev collocation method [26, 49] with $\mathcal{N}_x = 100$ degrees of freedom in each direction. This means the truth

CP tensor with $\lambda = 1$ (0.2308)

CP tensor with $\lambda = 10$ (0.2025)

CP tensor with $\lambda = 35$ (0.1921)

CP tensor with $\lambda = 100$ (0.1638)

CP tensor with $\lambda = 150$ (0.2881) practical regularization CP tensor (0.1296)

**Fig. 2** Recovered color image with relative reconstruction error norms provided in the bracket

approximation has dimension $\mathcal{N} = 10000 (\mathcal{N} = \mathcal{N}_x^2)$. The parameter domain $\mathcal{D}$ for $(\mu_1, \mu_2)$ is taken to be $[-0.99, 0.99] \times [-0.99, 0.99]$. For the parameter sample set, we discretize $\mathcal{D}$ using on a $9 \times 9$ cartesian grid, thus the size of training parameters is 81. The testing set $\Xi_{\text{test}}$ contains another 10 random samples in $\mathcal{D}$. The resulting tensor $\mathcal{A}$ is of dimension $100 \times 100 \times 81$. Given an initial value of the rank $R_0 = 50$ and tolerance $\varepsilon = 10^{-2}$, we run our algorithm on $\mathcal{A}$ at first, then we sort the rescaling coefficients $\{\alpha_r\}_{r=1}^{R_0}$ in descending order while discarding the coefficients below $\varepsilon\alpha_{max}$, where $\alpha_{max}$ is the maximum rescaling coefficients. Assuming that the number of the coefficients we keep is $R$ which is much
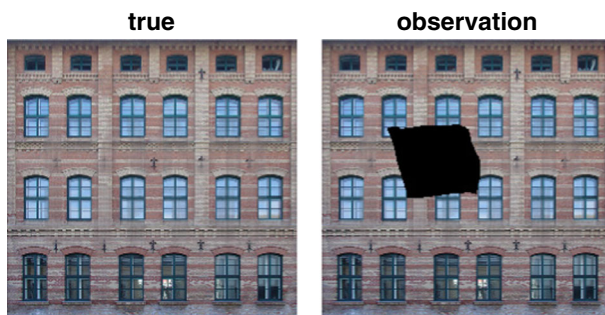
**Fig. 3** True image (left) and observation (right)



CP tensor $(0.0950)$         practical regularization
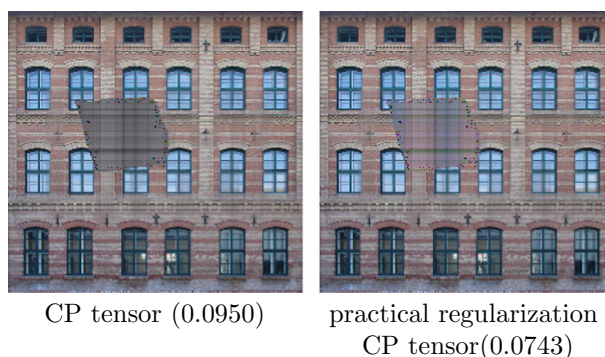                             CP tensor$(0.0743)$

**Fig. 4** Recovered color image with relative reconstruction error norms provided in the bracket

smaller than $\mathcal{N}$, then we run the regularized alternating optimization method with rank $R$ to approximate

$$\mathcal{A} \approx \sum_{r=1}^{R} \alpha_r \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r, \tag{4.2}$$

and build up the reduced bases $\{\phi_r\}_{r=1}^{R}$ by orthonormalizing $\{\widehat{\phi}_r\}_{r=1}^{R}$, where $\widehat{\phi}_r$ is created by vectorizing $\mathbf{x}_r \circ \mathbf{y}_r$.

### 4.2.1 Computational Performance

Given initial value of $\lambda = 10$, our proposed approach chooses $R = 20$ reduced bases, while the classical CP tensor chooses $R = 7$ reduced bases with the same initialization. The absolute value of the coefficients $\alpha$ is shown in Fig. 5. We can see that original CP tensor does more rank reduction than practical regularization CP tensor. However, it is noted that, in (3.9), larger regularization parameter $\lambda$ always gives more sparse solution, which doesn't mean solving this inverse problem more accurately. Thus, appropriate selection of $\lambda$ is crucial for solving (3.9), which is not easy if the prior knowledge or experience is not enough. In addition, in this model reduction example, we found that the sparsity of $\alpha$ is very sensitive to the choice of initial value of $\lambda$, that is, for classical CP tensor, much less(more) components of $\alpha$ will be close to zeros if we increase(decrease) $\lambda$ a little bit. To reduce the sensitivity caused by the determination of the initial value of $\lambda$, our practical regularization CP tensor includes
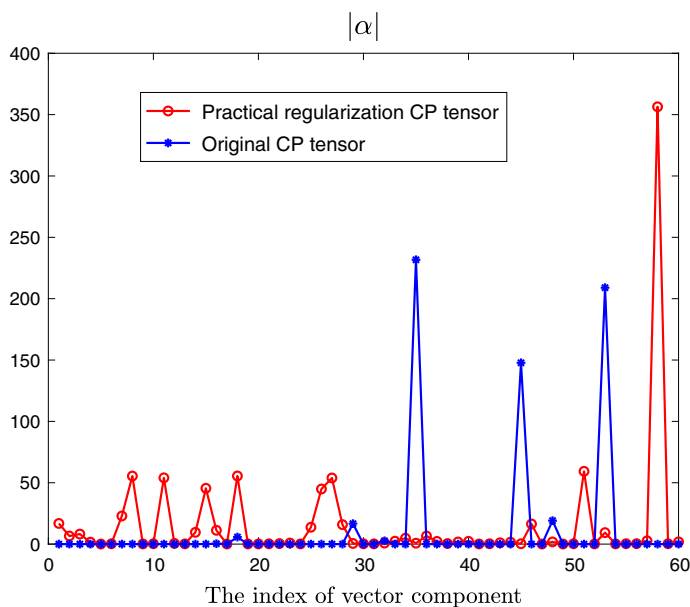
**Fig. 5** Value of $|\alpha|$: the comparison between practical regularization CP tensor and original CP tensor

the automatic regularization parameter selection, which is more robust than the conventional CP tensor since it doesn't rely on the empirical choice of $\lambda$.

To gain the understanding of the quality of the reduced bases our proposed approach, the number of reduced bases constructed by the POD is also set to be 20 for comparative purposes. The algorithmic accuracy is evaluated by approximating the $u(x, \mu_1, \mu_2)$ via reduced bases, where $(\mu_1, \mu_2) \in \Xi_{\text{test}}$ and the error is measured by $\ell^2$ norm. Fig. 6 displays the approximation of the solution at two parameters drawn from the testing set for the practical regularization CP tensor and the POD, since the 7 bases obtained by original CP tensor are too less to capture the main pattern of the solution. We observe that our proposed algorithm faithfully captures the feature of the solution, although the error is large compared to the POD. The performance of approximating the solutions at all the parameters in the testing set for our algorithm, canonical CP tensor and the POD is provided in Fig. 7. Table 1 further quantifies the range of the approximation results as well as the computational time. Although our practical regularization CP tensor is more expensive, the approximation quality of this method is better than original CP tensor since the automatic regularization parameter selection helps to reduce dependence on the initial guess of the regularization parameter. However, it can clearly be seen that POD is much faster and more accurate than the two CP tensor methods. Compared with POD, the main attraction of CP tensor methods is the low-cost storage, which is detailed in the following subsection.

### 4.2.2 A Note on Compression

For this model reduction problem, although both practical regularization CP tensor and CP tensor provide a more parsimonious representation of the data than the POD, comparing the compression ratios between the CP tensor techniques and the POD illustrates the difference and the benefit of the CP tensor techniques. For a rank $R = 20$ tensor of dimension $\mathcal{A} =$
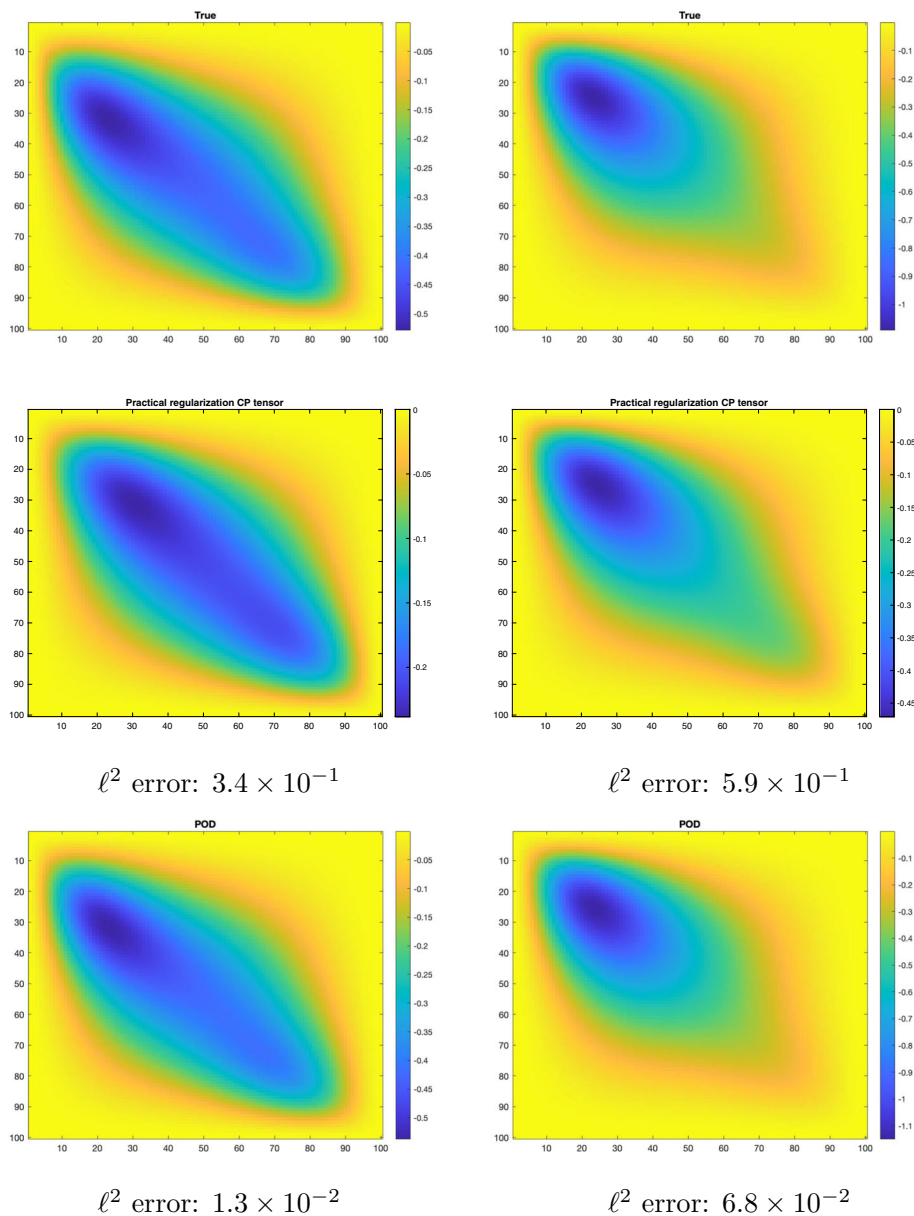
**Fig. 6** The approximation of the solution of Two-dimensional diffusion equation at two parameters in the testing set: $\mu_1 = -0.9193$, $\mu_2 = 0.6913$(left) and $\mu_1 = -0.8986$, $\mu_2 = -0.7977$(right)
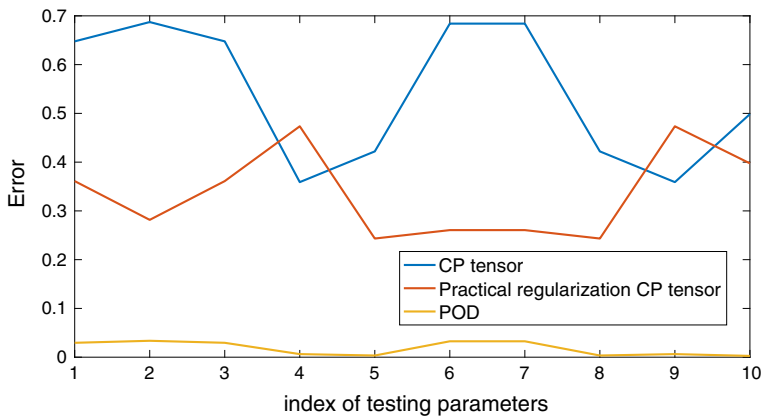
**Fig. 7** The approximation of $u(x, \mu_1, \mu_2)$ with $(\mu_1, \mu_2) \in \Xi_{\text{test}}$: CP tensor, practical regularization CP tensor and POD

**Table 1** Summary of the computational results for the model reduction problem

|              | CP tensor     | practical regularization CP tensor | POD              |
|--------------|---------------|-------------------------------------|------------------|
| Error        | [0.36, 0.83]  | [0.24, 0.58]                        | [0.003, 0.068]   |
| Number of bases | 7          | 20                                  | 20               |
| Time (s)     | 70.4          | 119.0                               | 0.03             |

The errors correspond to the approximation of $u(x, \mu_1, \mu_2)$ with $(\mu_1, \mu_2) \in \Xi_{\text{test}}$

$100 \times 100 \times 81$, the compression, the compression ratios are

$$\mathcal{C}_{\text{POD}} = \frac{I \cdot J \cdot K}{R \cdot (I \cdot J + K + 1)} = \frac{100^2 \cdot 81}{20 \cdot (100^2 + 81 + 1)} \approx 4.02, \tag{4.3}$$

$$\mathcal{C}_{\text{CP}} = \frac{I \cdot J \cdot K}{R \cdot (I+J + K + 1)} = \frac{100^2 \cdot 81}{20 \cdot (100 + 100 + 81 + 1)} \approx 143.62. \tag{4.4}$$

Notice that the POD requires the tensor to be reshaped in some direction. The comparison illustrates the striking difference between the compression ratios. It is worth mentioning that the CP tensor approaches requires much less memory to approximate the data. This can be of importance if the online stage (approximating the data) is in limited storage situations and that the accuracy requirement is not high.

## 5 Conclusion

In this paper, we have presented a new low-rank CP tensor completion algorithm by combining the flexible hybrid method and the CP tensor completion. A key advantage of this method is that the regularization parameter can be easily and automatically estimated during the iterative process, which substantially reduces the difficulty of initializing the regularization parameter and improves the robustness of the algorithm. In addition to memory savings, our proposed approach demonstrates outstanding performance on the model reduction example, compared to the POD. Moreover, our image recovery experiments show that our algorithm has a practical advantage in capturing more details in image reconstruction over the conventional

CP tensor due to a more optimal choice of the regularization parameter. In our future outlook, we will extend this hybrid approach in a tensor based total variation formulation for denoising and deblurring multi-channel images and videos.

**Data availability**  Enquiries about data availability should be directed to the authors.

## Declarations

**Conflict of interest**  The authors have not disclosed any competing interests.

## Appendix A: Proximal Gradient

Recall $g(\sigma) = \lambda \parallel \sigma \parallel_1$ is convex and non-differentiable. The function g can be turn into a proximal operator to find its minimum using the definition [41, 46] below:

**Definition A.1**  Given a proper closed convex function f: $\mathbb{R}^n \longrightarrow \mathbb{R} \bigcup \infty$, the proximal operator scaled by $\delta > 0$, is a mapping from $\mathbb{R}^n \longrightarrow \mathbb{R}$ defined by

$$\mathbf{prox}_{\delta g}(v) := \operatorname*{argmin}_{y \in \mathbb{R}^n} \left( g(y) + \frac{1}{2\delta} \parallel y - v \parallel^2 \right).$$

Then the proximal operator for $g(\sigma)$ is,

$$prox_g(v) = \operatorname*{argmin}_{\sigma} \left( g(\sigma) + \frac{1}{2\delta} \parallel \sigma - v \parallel_2^2 \right)$$

$$= \operatorname*{argmin}_{\sigma} \left( \lambda \parallel \sigma \parallel_1 + \frac{1}{2\delta} \parallel \sigma - v \parallel_2^2 \right)$$

$$= \operatorname*{argmin}_{\sigma} \left( \lambda \sum_{i=1}^{n} \mid \sigma_i \mid + \frac{1}{2\delta} \sum_{i=1}^{n} (\sigma_i - v)^2 \right)$$

For $\sigma = (\sigma_1, ..., \sigma_n)$

$$(prox_{\delta g}(v)_i)_i = \operatorname*{argmin}_{\sigma_i} \left( \lambda \mid \sigma_i \mid + \frac{1}{2\delta} (\sigma_i - v_i)^2 \right)$$

$$= \underset{\sigma_i}{\mathrm{argmin}} \begin{cases} \left(\lambda - \frac{v_i}{\delta}\right)\sigma_i + \frac{1}{2\delta}\sigma_i^2, \sigma_i > 0 \\ -\left(\lambda + \frac{v_i}{\delta}\right)\sigma_i + \frac{1}{2\delta}\sigma_i^2, \sigma_i < 0 \end{cases}$$

Hence,

$$prox_g(v)_i = \begin{cases} 0, & \mid v_i \mid \leq \lambda \\ v_i - \lambda sign(v_i), & \mid v_i \mid > \lambda \end{cases}$$

with $\delta = 1$

# References

1. Acar, E., Çamtepe, S.A., Krishnamoorthy, M.S., Yener, B.: Modeling and multiway analysis of chatroom tensors, ISI'05. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/11427995_21
2. Acar, E., Dunlavy, D.M., Kolda, T.G.: A scalable optimization approach for fitting canonical tensor decompositions. J. Chemom. **25**, 67–86 (2011). https://doi.org/10.1002/cem.1335
3. Acar, E., Dunlavy, D.M., Kolda, T.G., Mørup, M.: Scalable tensor factorizations for incomplete data. Chemom. Intell. Lab. Syst. **106**, 41–56 (2011). https://doi.org/10.1016/j.chemolab.2010.08.004
4. Andersson, C.A., Bro, R.: Improving the speed of multi-way algorithms: Part i. tucker3. Chemom. Intell. Lab. Syst. **42**, 93–103 (1998). https://doi.org/10.1016/S0169-7439(98)00010-0
5. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)
6. Beckmann, C., Smith, S.: Tensorial extensions of independent component analysis for multisubject fmri analysis. NeuroImage **25**, 294–311 (2005). https://doi.org/10.1016/j.neuroimage.2004.10.043
7. Beylkin, G., Mohlenkamp, M.J.: Algorithms for numerical analysis in high dimensions. SIAM J. Sci. Comput. **26**, 2133–2159 (2005)
8. Binev, P., Cohen, A., Dahmen, W., DeVore, R., Petrova, G., Wojtaszczyk, P.: Convergence rates for greedy algorithms in reduced basis methods. SIAM J. Math. Anal. **43**, 1457–1472 (2011)
9. Biswas, S.K., Milanfar, P.: Linear support tensor machine with lsk channels: pedestrian detection in thermal infrared images. IEEE Trans. Image Process. **26**, 4229–4242 (2017). https://doi.org/10.1109/TIP.2017.2705426
10. Björck, Å.: A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. BIT Numer. Math. **28**, 659–670 (1988)
11. Boelens, A.M., Venturi, D., Tartakovsky, D.M.: Tensor methods for the boltzmann-bgk equation. J. Comput. Phys. **421**, 109744 (2020). https://doi.org/10.1016/j.jcp.2020.109744
12. Bro, R.: Parafac tutorial and applications. Chemom. Intell. Lab. Syst. **38**, 149–171 (1997). https://doi.org/10.1016/S0169-7439(97)00032-4
13. Candes, E.J., Tao, T.: The power of convex relaxation: near-optimal matrix completion. arXiv:abs/0903.1476 (2009)
14. Chadwick, P.: Principles of continuum mechanics by m. n. l. narasimhan. John Wiley & Sons. 1993. 567 pp. isbn 0 471 54000 5. £48.95. Journal of Fluid Mechanics, 293, 404–404 (1995). https://doi.org/10.1017/S0022112095211765
15. Chung, J., Nagy, J.G., O'leary, D.P.: A weighted gcv method for lanczos hybrid regularization. Electr. Trans. Numer. Anal. **28**, 2008 (2008)
16. de Silva, V., Lim, L.-H.: Tensor rank and the ill-posedness of the best low-rank approximation problem. arXiv Mathematics e-prints. arXiv:math/0607647 (2006)
17. Erichson, N.B., Manohar, K., Brunton, S.L., Kutz, J.N.: Randomized cp tensor decomposition. Mach. Learn. Sci. Technol. **1**, 025012 (2020)
18. Figueiredo, M.A.T., Bioucas-Dias, J.M., Nowak, R.D.: Majorization-minimization algorithms for wavelet-based image restoration. IEEE Trans. Image Process. **16**, 2980–2991 (2007). https://doi.org/10.1109/TIP.2007.909318
19. Gazzola, S.: Flexible krylov methods for lp regularization
20. Gazzola, S., Sabaté Landman, M.: Krylov methods for inverse problems: surveying classical, and introducing new, algorithmic approaches. GAMM-Mitteilungen **43**, e202000017 (2020)

21. Geng, L., Nie, X., Niu, S., Yin, Y., Lin, J.: Structural compact core tensor dictionary learning for multispectral remote sensing image deblurring. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 2865–2869 (2018). https://doi.org/10.1109/ICIP.2018.8451531

22. Ghassemi, M., Shakeri, Z., Sarwate, A.D., Bajwa, W.U.: STARK: Structured Dictionary Learning Through Rank-one Tensor Recovery. arXiv:1711.04887 (2017)

23. Giryes, R., Elad, M., Eldar, Y.C.: The projected gsure for automatic parameter tuning in iterative shrinkage methods. Appl. Comput. Harmon. Anal. **30**, 407–422 (2011)

24. Gorodnitsky, I., Rao, B.: A new iterative weighted norm minimization algorithm and its applications. In: [1992] IEEE Sixth SP Workshop on Statistical Signal and Array Processing, pp. 412–415. IEEE (1992)

25. Hansen, P.C.: Discrete inverse problems: insight and algorithms. In: SIAM (2010)

26. Hesthaven, J.S., Gottlieb, S., Gottlieb, D.: Spectral Methods for Time-Dependent Problems, vol. 21. Cambridge University Press, Cambridge (2007)

27. Hou, M.: Tensor-based regression models and applications (2017)

28. Knuth, D.E.: The Art of Computer Programming, Seminumerical Algorithms. Addison-Wesley Longman Publishing Co., Inc, New York (1997)

29. Kolda, T.G., Bader, B.W., Kenny, J.P.: Higher-order web link analysis using multilinear algebra. In: ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining, pp. 242–249 (2005). https://doi.org/10.1109/ICDM.2005.77

30. Kruskal, J.: Three-way arrays: rank and uniqueness of trilinear decompositions, with applications to arithmetic complexity and statistics. Linear Algebra Appl. **18**, 95–138 (1977)

31. Li, N., Kindermann, S., Navasca, C.: Some convergence results on the regularized alternating least-squares method for tensor decomposition. arXiv:abs/1109.3831 (2011)

32. Li, N., Kindermann, S., Navasca, C.: Some convergence results on the regularized alternating least-squares method for tensor decomposition. Linear Algebra Appl. **438**, 796–812 (2013)

33. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 208–220 (2013). https://doi.org/10.1109/TPAMI.2012.39

34. Lorentz, G.G., Golitschek, M.V., Makovoz, Y.: Constructive Approximation: Advanced Problems, vol. 304. Springer, Berlin (1996)

35. Makantasis, K., Doulamis, A.D., Doulamis, N.D., Nikitakis, A.: Tensor-based classification models for hyperspectral data analysis. IEEE Trans. Geosci. Remote Sens. **56**, 6884–6898 (2018). https://doi.org/10.1109/TGRS.2018.2845450

36. Morozov, V.A.: On the solution of functional equations by the method of regularization. In: Doklady Akademii Nauk, vol. 167, pp. 510–512. Russian Academy of Sciences (1966)

37. Navasca, C., De Lathauwer, L., Kindermann, S.: reducing technique for tensor decomposition. In: 2008 16th European Signal Processing Conference, pp. 1–5. IEEE (2008)

38. Nouy, A.: Low-rank tensor methods for model order reduction. arXiv:1511.01555 (2015)

39. Ohlberger, M., Smetana, K.: Approximation of skewed interfaces with tensor-based model reduction procedures: application to the reduced basis hierarchical model reduction approach. J. Comput. Phys. **321**, 1185–1205 (2016)

40. O'Leary, D.P., Simmons, J.A.: A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. SIAM J. Sci. Stat. Comput. **2**, 474–489 (1981)

41. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. **1**, 123–231 (2014)

42. Pinkus, A.: N-widths in Approximation Theory, vol. 7. Springer, Berlin (2012)

43. Rodrıguez, P., Wohlberg, B.: An efficient algorithm for sparse representations with lp data fidelity term. In: Proceedings of 4th IEEE Andean Technical Conference (ANDESCON) (2008)

44. Rozza, G., Huynh, D.B.P., Patera, A.T.: Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Archiv. Comput. Methods Eng. **15**, 1 (2007)

45. Sanogo, F., Navasca, C.: Tensor completion via the cp decomposition. In: 2018 52nd Asilomar Conference on Signals, Systems, and Computers, pp. 845–849 (2018). https://doi.org/10.1109/ACSSC.2018.8645405

46. Selesnick, I.: Sparse regularization via convex analysis. IEEE Trans. Signal Process. **65**, 4481–4494 (2017). https://doi.org/10.1109/TSP.2017.2711501

47. Smilde, A., Bro, R., Geladi, P.: Multi-way analysis with applications in the chemical sciences (2004)

48. Stein, C.M.: Estimation of the mean of a multivariate normal distribution. Ann. Stat. **9**, 1135–1151 (1981)

49. Trefethen, L.N.: Spectral methods in MATLAB. In: SIAM (2000)

50. Uschmajew, A.: Local convergence of the alternating least squares algorithm for canonical tensor approximation. SIAM J. Matrix Anal. Appl. **33**, 639–652 (2012). https://doi.org/10.1137/110843587

51. Vogel, C.R.: Computational methods for inverse problems. In: SIAM (2002)

52. Walczak, B., Massart, D.: Dealing with missing data: part i. Chemom. Intell. Lab. Syst. **58**, 15–27 (2001). https://doi.org/10.1016/S0169-7439(01)00131-9

53. Wang, X., Navasca, C.: Adaptive low rank approximation of tensors. In: Proceedings of the IEEE International Conference on Computer Vision Workshop (ICCVW, Santiago, Chile, 2015)
54. Wang, X., Navasca, C.: Low-rank approximation of tensors via sparse optimization. Numer. Linear Algebra Appl. **25**, 2183–2202 (2018). https://doi.org/10.1002/andp.19053221004
55. Xu, X., Wu, Q., Wang, S., Liu, J., Sun, J., Cichocki, A.: Whole brain fmri pattern analysis based on tensor neural network. IEEE Access **6**, 29297–29305 (2018). https://doi.org/10.1109/ACCESS.2018.2815770
56. Zhang, J.: Design and Application of Tensor Decompositions to Problems in Model and Image Compression and Analysis, PhD thesis, Tufts University (2017)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.